

A semi-automatic categorization in Semantic Wiki Pages

Ícaro Rafael Medeiros¹, Ig Ibert Bittencourt^{1,2}, Evandro Costa^{1,2}

¹Computation Institute -Federal University of Alagoas
Campus A. C. Simões, BR 104 -Norte, Km 97, C. Universitária, Maceió, AL, Brazil

²Federal University of Campina Grande
Rua Aprígio Veloso, 882, Bodocongó, 58109-900, Campina Grande, PB, Brazil
icaro.medeiros@gmail.com, ebc@fapeal.br, ibert@dsc.ufcg.edu.br

Abstract

Ontology engineering is a hard task, but in collaborative environments like Wikis, domain experts and knowledge engineers can work together to create ontologies using semantic metadata. This article introduces the basis of a recommendation system for semantic annotation (more specifically categorization) to users on pages that do not have annotated category on Semantic Wikis. It will be employed in an educational environment. This will be reached with basis on the extraction of information from pages with similar content, through mining of terms in the text, using a machine learning approach.

KEY WORDS: Ontology engineering, semantic annotation, semantic wiki, text categorization.

1 Introduction

Nowadays, the Web is facing a time of transition, when many researchers have been involved in organizing the content of pages and embedding their code to be read by computers, so that usual tasks as navigation and search will be improved. This community effort will make complex queries involving the extraction of information from several sites a usual thing.

What is more, Web is huge, but not smart enough to easily integrate all of those numerous pieces of information that a user really needs (Devedzic, 2006). Most of its content today is designed for humans to read, not for computer programs to manipulate meaningfully (Berners-Lee *et al.*, 2001). A new generation for the service, called Semantic Web, is being designed to change that pattern. According to Boley and Wagner (2001), this new-generation Web tries to represent information so that it can be used by machines not just for display purposes, but for automation, integration, and reuse across applications.

The main foundations for the construction of the Semantic Web are ontologies. If knowledge is represented through them, the inference, sharing and reuse of knowledge become easier. Roughly speaking, ontology of

a certain domain is about terminology (domain vocabulary), all essential concepts in the domain, their classification, their taxonomy, their relations (including all important hierarchies and constraints), and about domain axioms.

Formally speaking, to someone who wants to discuss about topics in a domain D using a language L , an ontology provides a catalog of the types of things assumed to exist in D ; the types in the ontology are represented in terms of concepts, relations, and predicates of L (Devedzic, 2006).

The current language used for formal representation of ontologies is the OWL (Ontology Web Language), standardized by the W3C, in its initiative to Semantic Web (Activity, 2001)¹.

Nevertheless, one of the problems in the creation of ontologies is the need of technical knowledge for their creation and maintenance. Despite the fact that OWL is in text format, based on the known XML, for non-technical users it becomes an obstacle. But, according to Shadbolt *et al.* (2006), the ontologies that will furnish the semantics for the Semantic Web must be developed, managed, and endorsed by practice communities.

Therefore, Semantic Wikis (which are built in collaborative communities) can support the ontology engineering process, where domain experts and knowledge

¹ For a more comprehensive coverage on the field of ontologies, see Staab and Studer (2004).

engineers work together to create a formal ontology (Schaffert, 2006). In such wikis, users collaboratively build semantic-enriched pages with annotations, thus automatically creating ontologies in a wide variety of fields.

Wiki systems are Web applications that enable multiple users to build pages collaboratively and quickly. The resulting pages are called Wikis. One of the greatest examples of this method is the free encyclopedia online Wikipedia². Semantic Wikis use semantic annotations such as categorization of pages, typed links and page attributes to offer some facilities for navigation, structure and search, allowing complex queries and intelligent visualization.

However, not all users use such metadata when they create their pages. There is no culture of embedding semantic annotation in the documents and also users often lack the perception of where and when to put them in the pages. For this reason, some Semantic Web facilities are unexplored due to such failure in metadata creation.

This is the reason why we propose a page classification system in a Semantic Wiki, which will be embedded in GraW³ (an educational environment) (Silva *et al.*, 2006). In this system, through text categorization methods, it will be suggested to users (learners) categories for the page they are editing. This way, they tend to evaluate the classification made by the software, improving its algorithm. Figure 1 shows an example of GraW, which represents the community of Software Engineering. It is possible to notice the quick access options like communities, colleagues, documents, etc.

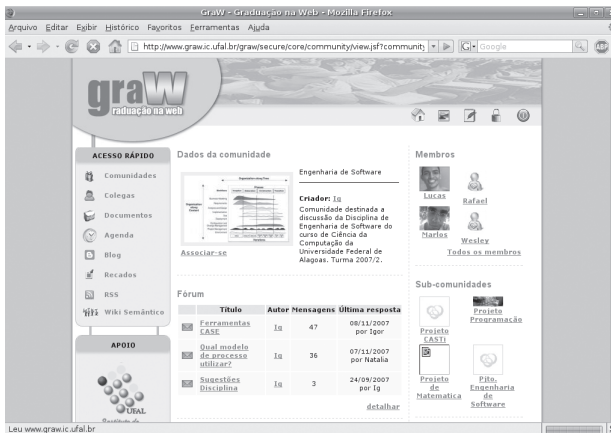


Figure 1. Engineering Software Community in GraW.

Thus, students will create ontologies collaboratively, and this activity increases their learning motivations (Damjanovic *et al.*, 2003). For this reason,

the proposed solution will be added to a learning environment.

Some works stated the advantages of Semantic Web Based Education (SWBE), like Devedzic (2006), which claims that the driving force for SWBE is not the technology but the strong motivation to fulfill the learner's growing needs in more comfortable ways, enabled by technology.

2 Analysis of text categorization algorithms

Text categorization (TC), also known as text classification, is the task of automatically sorting a set of documents into categories from a predefined set. TC may be formalized as the task of approximating the unknown target function $\Phi: D \times C \rightarrow \{T, F\}$ (that describes how documents ought to be classified, according to a supposedly authoritative expert) by means of a function $\Phi: D \times C \rightarrow \{T, F\}$ called the classifier, where $C = \{c_1, \dots, c_{|C|}\}$ is a predefined set of categories and D is a set of documents. If $\Phi(d_j, c_i) = T$, then d_j is called a positive example (or a member) of c_i , while if $\Phi(d_j, c_i) = F$ is a negative example of c_i (Sebastiani, 2005).

According to Sebastiani (2002), given $d_j \in D$ a system might simply rank the categories in $C = \{c_1, \dots, c_{|C|}\}$ according to their estimated appropriateness to d_j , without taking any "hard" decision on any of them. This ranked list could be of great help to a human expert in charge of making the final categorization decision.

This method is used here, where the classified categories will be displayed to users in order to decide which of them and in what order they are going to be used to file the document. The literature refers to this method as semi-automatic interactive classification.

Furthermore, text categorization methods using machine learning create an inductive process that, by observing the characteristics of a set of documents preclassified under c_i , gleans the characteristics that a new unseen document should have in order to belong to c_i (Sebastiani, 2005).

In order to create the classifier, the system must have a set of documents pre-classified by a domain expert. This set is called initial *corpus* $\Omega = \{d_1, \dots, d_{|\Omega|}\}$ in D where d_j are documents under C . Prior to this construction, Ω is split in two sets, but not necessarily of equal size (Sebastiani, 2002):

² The online encyclopedia Wikipedia can be seen at <http://www.wikipedia.org>.

³ GraW can be accessed at <http://www.graw.ic.ufal.br>.

- A training (-and-validation) set $TV = \{d_1, \dots, d_{|TV|}\}$. The classifier is built by observing the characteristics of these documents.
- A test set $Te: Te = \{d_1, \dots, d_{|Te|}\}$. It is used for testing the effectiveness of the classifiers. Classifiers get each $d_j \in Te$ and their decisions are compared with the expert decisions, and a measure of classification effectiveness is based on how often this comparison matches.

This is called the train-and-test approach (Sebastiani, 2002). The decision on what pages to use for the initial corpus and how many factors are required to achieve a high precision classifier.

Beyond these preclassified documents, the system will use ontologies to build the classifier. With user's information about how to classify the document in a better way, categorization methods are refined.

2.1 Text categorization in Wiki pages

Text categorization in Web pages, and more specifically Wikis, has some peculiarities that should be considered. Some Wiki Systems approaches available on the Internet pointed out that the first(s) term(s) of pages might define with good accuracy in which category they would be filed in.

To illustrate this, tests were made in *Artificial Intelligence* and *Automata Theory* Wikipedia in Portuguese⁴. In these, the first sentences are “Artificial intelligence is an area of research in computer science...” and “Automata Theory is a branch of computer science...”, respectively. Therefore, the term *computer science* determines the category of these pages (areas in computer science). As you might notice, the result is very intuitive.

This information is more relevant than, for example, word frequency (the most commonly used feature extraction) (Mahinovs and Tiwari, 2007). Russell and Norvig (2004) state that each document could be represented in a vector of frequency of each term. However, Sebastiani (2005) uses the weight terminology when states that a text d_j is usually represented as a vector of term weights $d_j = \{w_{1j}, \dots, w_{\tau j}\}$ where τ (dictionary) is a set of terms (sometimes called *features*) that occur at least once in at least one document of D , and $0 \leq w_{kj} \leq 1$ quantifies the importance of t_k in characterizing the semantics of d_j .

It is proposed, then, a system of importance levels of terms according to the way they occur, whose weight is

computed by taking into consideration few classes. So we are going to associate values here named *importance factors* (IFs), represented in decreasing order set $IF = \{f_1, \dots, f_n\}$. These factors are going to be used to evaluate the term weights, and their classes are described below, in its rank of relevance, for a set IF with $n=4$. These considerations make our proposal different from more general approaches of TC.

- Terms in the beginning of the page (f_1): As mentioned before, there are strong clues pointing that terms in the first sentences provide important information sources for categorization. If this term appears also frequently in other parts of the text, its importance grows;
 - Multiple occurrences of a term (f_2, f_3, f_4):
 - If there is a hyperlink and the Wiki to which it points exists (f_2).
 - If there is a hyperlink, but the Wiki to which it points does not exist (f_3).
 - There is no hyperlink (f_4).

$$w_{kj} = f_1 * o_{k1} + \dots + f_n * o_{kn} = \sum_{m=1}^n f_m * o_{km} \quad (\text{Equation 1})$$

Equation 1 expresses weight of a term t_k in a document d_j , where f_1, \dots, f_n are the importance factors of each class $cl = 1, \dots, n$ and o_{km} represents the number of occurrences of the term t_k under the category cl . Moreover, the weights resulting from w_{kj} are often normalized by the cosine normalization, in order to make weights fall in the $[0, 1]$ interval and documents be represented by vectors of equal length. This operation is given by Equation 2, where the denominator is the square root of sum of squares of the term weights of all the terms in the document.

$$w_{kj} = \frac{w_{kj}}{\sqrt{\sum_{s=1}^{\tau} w_{sj}^2}} \quad (\text{Equation 2})$$

3 Semantic Wiki architecture

For creating the Wiki system which is going to be attached to GraW, an existing application, IkeWiki (Schaffert, 2006), will be reused. This tool, which is open source (allowing to be extended in an easy way), has many desirable features that made us choose it at the expense of other solutions like SemanticMediaWiki (Krotzsch *et al.*, 2006) and PlatypusWiki (Campanini *et al.*, 2004):

⁴ These pages can be accessed via the following URLs: http://pt.Wikipedia.org/wiki/ntelig%C3%Aancia_artificial and http://pt.Wikipedia.org/wiki/Teoria_de_Aut%C3%B4matos.

- It allows immediate exploitation of semantic annotations for enhanced editing, presentation, navigation, and searching, even if the knowledge base is not yet fully formalized;
- It has an interactive *WYSIWYG* (What You See is What You Get) editor, using AJAX technology, also supporting semantic annotations;
- It supports Wikipedia syntax, allowing users to import existing content from Wikipedia into IkeWiki (e.g. via simple copy and paste) and begin with semantic annotations straight away;
- It is purely based on existing Semantic Web standards like RDF and OWL, to be able to exchange data with other applications (e.g. ontology editors, Semantic Web services, other Wikis);
- It allows the formalization of knowledge from informal texts to formal ontologies. Also, this means that parts of the knowledge base might be more formalized than others, and that formal knowledge is in constant evolution;
- Unlike most other Semantic Wikis, IkeWiki supports reasoning (using OWL-RDFS) on the knowledge base, which allows to derive knowledge that is not explicit in the data and is thus the true power of Semantic Web technology (Schaffert, 2006).

IkeWiki is implemented as a Java Web application using a layered architecture, where data is stored in a Postgres database. When a resource is requested, the XML page content and related RDF data are retrieved and combined in the *RenderingPipeline* into an enriched XML representation. This is then either offered as interchange format for other Web services or transformed into HTML for presentation in the user’s browser (Schaffert, 2006).

The extension presented is going to add a code layer to the edition part of IkeWiki, which is going to recommend categories to classify a document, following the steps below:

1. An extractor allocates the terms in their classes for the importance factor evaluation (see 2.1).
2. Term weights are calculated considering their stem (for example, the words *computer* and *computed* are seen as the same term), synonyms or similar cognates. Terms called stop words like (“the”, “of”, “or” and “to be”) are ignored.
3. The classifier is used, where information of the vector of term weights of document d_j are matched with the vectors dk in initial corpus Ω , creating the space vector model. The adequate category is found by relating the most relevant terms in $dk \in \Omega$ with the vector d_j .

Nearest neighbors are calculated by the shortest Euclidean distance, as shown in Equation 3, where $dist_{jk}$ is the distance between document d_j and $d_k \in \Omega$, j_m is the weight of a term t_m in d_j , k_m the weight of the same term in d_k and $m = |\tau|$

$$dist_{jk} = \sqrt{(j_1 - k_1)^2 + \dots + (j_m - k_m)^2} \tag{Equation 3}$$

For this comparison to work, all the vectors must be arranged so one coordinate represents the same term in all of them.

4. The categories of classified documents whose relevant terms are similar (according to the formula above) to those in d_j are returned, giving the user a list of recommended categories. According to the final answer of users on the task of categorization, the classifier is refined, because set D will have more classified documents, increasing its accuracy.

Figure 2 shows IkeWiki architecture plus the proposed extension, where three layers of code are shown - Extractor, Categorization and Persistence and their main classes.

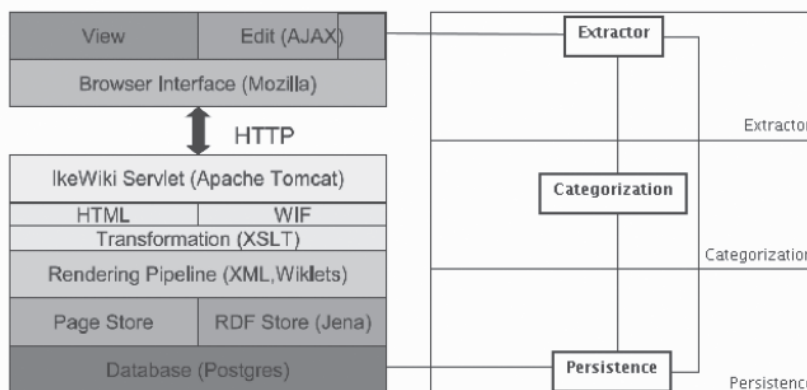


Figure 2. IkeWiki architecture (Schaffert, 2006) plus the proposed extension.

In the Extractor layer, the Extractor class accesses other classes with data about the syntax to extract information (terms) according to its type (as described on section 2.1). Also, it must have a connection with the Persistence layer, to verify the existing Wikis.

The categorization layer integrates the classes implementing the TC methods, dealing with the formulas for term weights, similarity of pages and generation of ranked lists of categories to documents. All this methods are accessed through the main class Categorization.

In the Persistence layer, whose main class is Persistence, there is access to the database, in a transparent way, which isolates this access from the rest of the code. IkeWiki and the extension mentioned will be embedded in GraW. An example of the algorithms and sequence of steps presented here is shown in the next section.

4 Illustrative scenario

The context of GraW provides an illustrative scenario about the use of the proposed human-system interaction. In GraW, actors in a professor’s role can propose new courses in the form of communities. The content of these courses can be available using the forum, while monitors can help students through chat and video-conference tools (de Almeida *et al.*, 2004).

Suppose we have, for example, Wikis for the communities of *Programming Language* and *Theory of Computation*, which are given hypothetical categorization as shown in Table 1:

Table 1. Categorized examples forming the training set Ω .

Pages	Categories
Programming Language	Programs – Languages – Code – Computation
Theory of Computation	Computation – Models – Problems

Table 2. Terms table.

1 – Languages	7 – Problem(s)	13 – Logics
2 – Program(ing)/Software	8 – Model(ing)	14 – Specification(s)
3 – Comput(er)ation	9 – Turing	15 – Requirement(s)
4 – Code(d)ing	10 – Resol(ve)ution	16 – Machine(s)
5 – Instruction(s)	11 – Intelligence	17 – State(s)
6 – Algorithm(s)	12 – Artificial	18 – Automat(on/a)

Suppose now that we intend to categorize the page *Automata Theory*. Gathering the terms found in this page with the terms in documents $d_j \in \Omega$ we have this list, as shown on Table 2 (only a few relevant terms were considered for each page).

So, a matrix is formed based on this table where the first row represents the normalized term weights of the document to be categorized and the other rows represent these weights in documents in Ω . The columns represent the terms weight in each document, where its coordinate number follows the ones presented in Table 2, i.e., the first column represents the weight of $k = \text{Language}$. This term weight, in document $j = \text{Automata Theory}$, for example, is given by

$$w_{kj} = \frac{\sum_{m=1}^4 f_m * o_{km}}{\sqrt{\sum_{s=1}^r w_{sj}^2}} = \frac{33}{\sqrt{26654}} = 0.2021309 .$$

In Table 3, it is shown the resulting matrix of weight (manually calculated according to the importance factor algorithm with texts in Wikipedia pages⁵), with a precision of 2 decimal places. Based on this matrix the distance is calculated. So, for $d_p = \text{Automata Theory}$, $d_q = \text{Theory of Computation}$ and $d_r = \text{Programming Language}$: Distance between d_p and $d_q = 0.6947954$.

Table 3. Term–document matrix (normalized) for d_p , d_q and d_r .

.20	0	.11	0	0	.01	0	.07	.12	.04	0	0	0	.01	0	.69	.45	.50
.04	.01	.79	0	.05	.11	.23	.43	.19	.11	0	0	0	0	.004	.27	0	.05
.48	.60	.32	.42	.07	.01	0	0	0	0	0	0	.03	.08	0	.35	0	0

As expected, *Automata Theory*, which in fact is an area that supports *Theory of Computation*, is the most similar document found. However, the difference between the two distance values is small, despite the fact that we know *Programming Language* is a very distinct subject when comparing with *Automata Theory*. This accuracy should be increased with more classified documents in the training set.

⁵ These pages are available through the following URLs: http://pt.Wikipedia.org/wiki/Teoria_de_Aut%C3%B4matos, http://pt.Wikipedia.org/wiki/Teoria_da_computa%C3%A7%C3%A3o and http://pt.Wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o.

Finally, the system returns a list containing the categories in closest documents in relation to d_p . Considering *Automata Theory*, these would be *Computation*, *Models* and *Problems*, because there is a relation between these terms and the categories. Nevertheless, a better classification would include *State Machine* as the first entry. For these cases in which the existing categories do not suit well for an optimum classification, users can add a new category, so the training set becomes smarter and some categorizations already done are reviewed, increasing its accuracy.

Users that suggest the “better” classification mentioned will be learners using GraW, and this activity of classifying pages might be helpful in the learning process, because while organizing pages, the structure of knowledge he/she gets with the e-learning environment also gets organized for him as the main keywords about the Wiki are reviewed. Also, when this task is made in a collaborative way it can lead students into constructive discussions about whether a page should or should not be classified under some categories. This kind of learning is called restructuring knowledge, when the relations between knowledge are considered to reconstruct the knowledge structure (Rumelhart and Norman, 1978).

Besides, it is better for students to find the pages well-organized in categories, so they do not need to follow related pages through links that might make them lose the focus. So, when a teacher advises his/her students to read all pages in Automata category in order to pass the exams he/she knows the students will not waste their time with irrelevant pages. As a side effect, students will learn about semantic metadata and its importance to enhance pages. Finally, ontologies can be an important learning object, and their creation depend on metadata available.

5 Related work

A lot of research has been done in text categorization, because achievements in this area can be very helpful for making semantic annotation widespread. The advent of Semantic Web concepts and the incredible amount of content in Wiki systems in pages like Wikipedia make Semantic Wiki powerful tools to test and use text mining and classification algorithms.

Some works try to take advantage of the huge content of Wikipedia for classifying tasks. Milne *et al.* (2006) propose the creation of thesauri (listing of words with similar, related, or opposite meanings) for specific domains based on Wikipedia’s content. In this project, called Wikisauri, comparing terms and semantic

annotations to those in a manually created domain-specific thesaurus demonstrates excellent coverage of domain terminology, and of synonym relations between terms. They have found that Wikipedia outperforms a professional thesaurus in supporting a domain-specific document collection.

Gabrilovich and Markovitch (2006) state that the bag of words, the traditional method used in TC (also used in this article), have a quite limited performance for more demanding tasks, such as those dealing with small categories or short documents. To empower machine learning techniques, an auxiliary text classifier is built that is capable of matching documents with the most relevant articles of Wikipedia. The conventional bag of words is enhanced with new features, which correspond to the concepts represented by these articles, which leads to substantially greater categorization accuracy.

In the TC field applied to education, we observed the work in Williams *et al.* (2003), which tries to automatically classify questions in an “ask-an-expert” system, in a mathematical context. This is very helpful considering the fact that these experts usually have experience in a strict subdomain of knowledge, so, it is useful that the questions they receive are in their fields of expertise.

Lui *et al.* (2007) use text classification to automatically classify online discussion analysis (this is usually used for teachers and researchers to analyze discussion forums helping in tasks like learner’s assessment and learner’s growth evaluation).

Educational environments also exploit the advantages of semantic metadata. Saini *et al.* (2006) propose a system that performs a classification of Learning Objects exploiting the prior knowledge encoded in the taxonomy, and then associates the proper metadata to these objects, in order to organize the learning resources into domain specific repositories, referred to as Learning Object Repositories.

Wikis can be very important for educational systems, as stated in Raman *et al.* (2005). In Muljadi *et al.* (2006), Semantic Wikis are proposed for the creation of lightweight knowledge management systems, in a case study towards a biology dictionary in Japanese.

Finally, Lange and Kohlhase (2006) use a Semantic Wiki to manage mathematical knowledge, where the target audience is composed of mathematicians developing new theories and scholars learning mathematics. With this system, it is possible to make complex queries like “retrieve all theorems about triangles for which a proof exists” or queries involving formulas and mathematical language.

6 Conclusion

This article has shown a proposal for semi-automatic TC of Wiki pages using machine learning techniques. Using this, users can evaluate the categorizations suggested, so the algorithms are refined, and semantic annotation is created. Finally, semantic-enhanced features of GraW Wiki system can be fully available.

As future work, the proposed system will be developed with the improvement of algorithms and tests in huge databases. After that, we will test the algorithms with Wikipedia articles about Computer Science and Informatics (about 1800 articles). Besides, some consideration made in Gabrilovich and Markovitch (2006) can be added to our system. Finally, we also plan to use the idea of considering semantic relations between pages, in addition of the relevant terms set.

References

- ACTIVITY, W.S. 2001. Semantic Web activity statement. Available at: <http://www.w3.org/2001/sw>. Accessed on: 10/15/2007 .
- BERNERS-LEE, T.; HENDLER, J. and LASILLA, O. 2001. The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 5(284):34-43.
- BOLEY, S.T.H. and WAGNER, G. 2001. Design rationale of RuleML: A markup language for semantic web rules. In: I.F. CRUZ; S. DECKER; J. EUZENAT and D.L. MCGUINNESS, (eds.), SEMANTIC WEB WORKING SYMPOSIUM, I, Stanford, California, USA, 2001. *Proceedings...* Stanford, Elsevier Science Publishers B. V., p. 381-402.
- CAMPANINI, S.E.; CASTAGNA, P. and TAZZOLI, R. 2004. Platypus wiki: a semantic wiki wiki web. In: SWAP : SEMANTIC WEB APPLICATIONS AND PERSPECTIVES, I, Ancona, Italy, 2004. *Proceedings...* Ancona, 2004. Available at: <http://semantieweb.deit.univpm.it/swap2004/cameraready/castagna.pdf>.
- DE ALMEIDA, H.O.; TENÓRIO, L.E.F.; DE BARROS COSTA, E.; BARBOSA, N.M.; BUBLITZ, F.M., and BARBOSA, A.A. 2004. Um Arcabouço de Software Livre baseado em Componentes para a Construção de Ambientes de Comunidades Virtuais de Aprendizagem na Web. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE'04), XV, Manaus, Amazonas, 2004. *Proceedings...* Manaus, vol. 15, p. 188-196.
- DAMJANOVIC, V.; GASEVIC, D. and DEVEDZIC, V. 2003. Ontology validation. In: INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY (CIT 2003), VI, Bhubaneswar, India, 2003. *Proceedings...* Bhubaneswar, p. 183-186.
- DEVEDZIC, V. 2006. *Semantic Web and Education (Integrated Series in Information Systems)*. 1st ed., New York, Springer-Verlag, 353 p.
- GABRILOVICH, E. and MARKOVITCH, S. 2006. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, XXI, Boston, MA, USA, 2006. *Proceedings...* Boston, AAAI press, p. 1301-1306.
- KRÖTZSCH, M.; VRANDEIC, D. and VÖLKELE, M. 2006. Semantic mediawiki. In: I. CRUZ; S. DECKER; D. ALLEMANG; C. PREIST; D. SCHWABE; P. MIKA; M. USCHOLD and L. AROYO (eds.). In: INTERNATIONAL SEMANTIC WEB CONFERENCE (ISWC06), V, Athens, 2006. *Lecture Notes in Computer Science*, vol. 4273. Berlin, Springer, p. 935-942.
- LANGE, C. and KOHLHASE, M. 2006. A semantic wiki for mathematical knowledge management. In: M. VÖLKELE and S. SCHAFFERT (eds.), WORKSHOP ON SEMANTIC WIKIS – FROM WIKI TO SEMANTICS, I, Budva, Montenegro, 2006. *Proceedings...* Budva, CEUR-WS.org, CEUR Available at: <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-206/>. Accessed on: 10/17/2007.
- LUI, A.K.-F.; LI, S.C. and CHOY, S.-O. 2007. An evaluation of automatic text categorization in online discussion analysis. In: INTERNATIONAL CONFERENCE ON ADVANCED LEARNING TECHNOLOGIES (ICALT), VII, Niigata, Japan, 2007. *Proceedings...* Niigata, IEEE Computer Society, p. 205-209.
- MAHINOV, A. and TIWARI, A. 2007. Text classification method review. In: R. ROY and D. BAXTER (eds.), *Decision Engineering Report Series*. Cranfield, Cranfield University, Available at: <https://aerade.cranfield.ac.uk/bitstream/1826/1860/1/mahinovs.pdf>. Accessed on: 10/18/2007.
- MILNE, D.; MEDELYAN, O. and WITTEN, I. H. 2006. Mining domain-specific thesauri from wikipedia: A case study. In: IEEE/WIC/ACM INTERNATIONAL CONFERENCE ON WEB INTELLIGENCE, V, Hong Kong, 2006. *Proceedings...* Washington, DC, USA, IEEE Computer Society, p. 442-448.
- MULJADI, H.; TAKEDA, H.; SHAKYA, A.; KAWAMOTO, S.; KOBAYASHI, S.; FUJIYAMA, A. and ANDO, K. 2006. Semantic wiki as a lightweight knowledge management system. In: R. MIZOGUCHI; Z. SHI and F. GIUNCHIGLIA (eds.), *Lecture Notes in Computer Science*. Berlin, Springer, vol. 4185, p. 65-71.
- RAMAN, M.; RYAN, T. and OLFMAN, L. 2005. Designing knowledge management systems for teaching and learning with wiki technology. *Journal of Information Systems Education*, 16(3):311-321.
- RUMELHART, D.E. and NORMAN, D.A. 1978. Accretion, tuning and restructuring: Three modes of learning. In: J. COTTON and R. KLATZKY (eds.), *Semantic Factors in Cognition*. Hilldale, Lawrence Erlbaum Associates, p. 37-54.
- RUSSELL, S. and NORVIG, P. 2004. *Inteligência Artificial*. 2nd ed., São Paulo, Elsevier/Campus, 1040 p.
- SAINI, P.S.; RONCHETTI, M. and SONA, D. 2006. Automatic generation of metadata for learning objects. In: IEEE INTERNATIONAL CONFERENCE ON ADVANCED LEARNING TECHNOLOGIES, VI, Kerkrade, The Netherlands, 2006. *Proceedings...* Washington, DC, USA, p. 275-279.
- SCHAFFERT, S. 2006. Ikwiki: A semantic wiki for collaborative knowledge management. In: WORKSHOPS ON ENABLING TECHNOLOGIES: INFRASTRUCTURE FOR COLLABORATIVE ENTERPRISES, XV, University of Manchester, Manchester, UK, 2006. *Proceedings...* Washington, DC, USA, IEEE Computer Society, p. 388-396.
- SEBASTIANI, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1-47.
- SEBASTIANI, F. 2005. Text categorization. In: A. ZANASI; (ed.), *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*. Southampton, WIT Press, p. 109-129.
- SHADBOLT, N.; BERNERS-LEE, T. and HALL, W. 2006. The

- semantic web revisited. *IEEE Intelligent Systems*, **21**(3):96-101.
- SILVA, C.; BITTENCOURT, I.I.; DE BARROS COSTA, E.; AGUIAR, M.; SIBALDO, M. and TADEU, M. 2006. Construção de comunidades virtuais de aprendizagem na web através do arcabouço de software livre arco. *REIC, Revista Eletrônica de Iniciação*, **4**:21-36.
- STAAB, S. and STUDER, R. (ed.). 2004. *Handbook on Ontologies*. 1st ed., New York, Springer-Verlag, 660 p.
- WILLIAMS, K.; CALVO, R.A. and BELL, D. 2003. Automatic categorization of questions for a mathematics education service. *In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, XI*, Sydney University, Sydney, Australia, 2003. *Proceedings...* Berkeley, International AIED Society, p. 81-87.

Submitted on January 29, 2008

Accepted on February 19, 2008